



Détection de différence de mise au point lors des prises de vues stéréoscopiques

Sergi Pujades, Frédéric Devernay

► To cite this version:

Sergi Pujades, Frédéric Devernay. Détection de différence de mise au point lors des prises de vues stéréoscopiques. Orasis, Congrès des jeunes chercheurs en vision par ordinateur, Jun 2013, Cluny, France. hal-00829397

HAL Id: hal-00829397

<https://hal.science/hal-00829397>

Submitted on 5 Jun 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Détection de différence de mise au point lors des prises de vues stéréoscopiques

Sergi Pujades
sergi.pujades-rocamora@inria.fr

Inria Grenoble

Frédéric Devernay
frederic.devernay@inria.fr

Résumé

La production d'images stéréoscopiques nécessite un rig stéréoscopique avec deux caméras parfaitement synchronisées et alignées. La plupart des imprécisions de ce montage peuvent être corrigées en direct ou en post-production. Par contre, une différence de distance de mise au point ou de profondeur de champ entre les caméras produira des dégradations irrécupérables dans les images. Dans cet article nous détectons des différences de mise au point entre les deux vues d'une paire stéréoscopique en quatre étapes. D'abord nous calculons une carte de disparité dense. Ensuite nous mesurons la netteté dans chaque image et nous comparons ces mesures. Après, avec des méthodes statistiques robustes, nous identifions les zones de l'image qui présentent des différences. Finalement, nous proposons une méthode de visualisation sur les images originales pour informer l'opérateur des problèmes, et lui donner des indices pour les résoudre.

Mots Clef

Cinéma stéréoscopique, flou, netteté, distance de mise au point, *depth from focus*, *depth from defocus*.

Abstract

Live-action stereoscopic content production requires a stereo rig with two cameras precisely matched and aligned. While most deviations from this perfect setup can be corrected either live or in post-production, a difference in the focus distance or focus range between the two cameras will lead to unrecoverable degradations of the stereoscopic footage. In this paper we detect focus mismatch between views of a stereoscopic pair in four steps. First, we compute a dense disparity map. Then, we use a measure to compare focus in both images. After this, we use robust statistics to find which images' zones have a different focus. Finally, to give useful feedback, we show the results on the original images and give hints on how to solve the focus mismatch.

Keywords

Stereoscopic cinema, Focus mismatch, Depth from focus, Depth from defocus.

1 Introduction

Dans cet article nous proposons une méthode pour détecter des différences de *mise au point* (MAP) dans des images

stéréoscopiques. Ces différences apparaissent quand la *distance de mise au point* (DMP) ou la *profondeur de champ* (PC) ne sont pas identiques pour les deux caméras.

La production d'images stéréoscopiques nécessite un rig stéréoscopique avec deux caméras parfaitement synchronisées et alignées. Ces réglages se font sur le lieu du tournage et prennent du temps. La plupart des imprécisions de ce montage peuvent être corrigées en direct ou en post-production. Par contre, une différence de DMP ou de PC entre les caméras produira des dégradations irrécupérables dans les images. Il est donc important de détecter ces incohérences rapidement pour que l'opérateur puisse les corriger sur le tournage.

Le lien entre la différence de MAP entre les deux images et la fatigue cérébrale ne fait pas l'unanimité. Les études se sont concentrées sur la visualisation d'une image nette avec une image floue [5, 3, 12, 14], mais les effets de deux images ayant la même PC et des DMP différentes n'ont pas été étudiés à notre connaissance. Dans la TV-3D et le cinéma 3D, la MAP joue aussi un rôle artistique, et la plupart des réalisateurs croient qu'une désynchronisation de MAP peut interférer dans l'effet relief [4].

Nous voulons détecter cette différence de MAP avec l'information provenant exclusivement des images capturées. Nous considérons que nous n'avons aucune information provenant des optiques : ni leur focale, ni leur DMP, ni la taille d'ouverture de leur iris. Bien qu'il existe des protocoles bien établis pour l'obtention de ces informations, toutes les optiques ne sont pas compatibles avec ces standards. Dans la plupart des cas les informations obtenues sont imprécises, puisque la calibration est faite en usine alors que les valeurs réelles dépendent de la position de l'optique par rapport au capteur de la caméra.

Mesurer le *flou dû à l'optique* (FO) à partir du *flou dans l'image* (FI) est un problème mal posé. Si nous avons une zone de couleur homogène dans la scène, nous ne pouvons pas distinguer une image nette d'une image floue car les deux seront fortement similaires. D'autre part, l'image nette d'un poster montrant une image floue apparaîtra floue. Dans notre cas nous ne voulons pas mesurer le FO : nous voulons détecter une *différence* entre les deux images, puisque c'est cette incohérence qui risque de perturber la perception stéréoscopique. Dans nos travaux, nous ne traitons pas non plus la problématique de la restauration d'une image floue pour la rendre nette. Notre but se limite à détecter s'il y a une différence entre la DMP et/ou la PC des

deux caméras. Nous imposons une contrainte de rapidité à notre méthode, afin d'être utilisable pendant le tournage. Nous voulons aussi être capables de donner de l'information au chef opérateur ou au réalisateur sur la nature du problème. Est-ce que la PC est la même pour les deux caméras ? Les deux caméras ont-elles la même DMP ? Ce sont des questions auxquelles nous voulons répondre.

Dans cet article, par manque de place, certaines figures ont dû être fortement réduites. Merci de se référer à la version électronique pour voir les détails.

2 Modèle de flou, différence de flou et le flou dans les images

2.1 Modèle de flou dû à l'optique

Le flou dans une image est généralement modélisé comme une convolution d'une image parfaitement nette avec une fonction de dispersion (PSF). Dans notre cas, nous supposons que cette fonction est en relation avec le FO, et nous ignorons les effets qui pourraient y contribuer de façon mineure (aberrations optiques, diffraction, etc).

La fonction de dispersion a suscité plusieurs modélisations dans la littérature [1, 16], bien que le modèle le plus utilisé soit celui d'une gaussienne 2D [7, 9, 15, 6] (la fonction de dispersion réelle dépend, entre autres, de la forme de l'iris). La taille du FO, σ , dépend de la distance du point 3D imagé au plan du diaphragme. On appellera cette distance la profondeur. Il ne faut pas la confondre avec la distance entre le point 3D imagé et le centre optique (point au centre du diaphragme). Dans notre cas, les surfaces de points 3D qui se retrouvent à la même profondeur ne forment pas des hémisphères mais des plans. Si on a deux caméras en configuration stéréoscopique, les axes optiques sont presque parallèles et orthogonaux à l'entraxe reliant les deux centres optiques. Dans ce cas, la profondeur est la même pour les deux caméras (les deux plans des diaphragmes coïncident) et la disparité stéréoscopique dépend simplement de la profondeur.

Une propriété importante démontrée par [10, 11] est que la taille de FO a une relation linéaire avec la disparité stéréoscopique, comme l'illustre la Fig. 1. L'intuition derrière cette relation est qu'une image prise avec une PC limitée, correspondant à un diaphragme donné et une DMP, peut être interprétée comme la moyenne d'une infinité d'images prises avec une PC infinie (un diaphragme infiniment petit réduit au centre optique), et toutes convergentes à la DMP. Cette infinité de centres optiques couvrent complètement la surface du diaphragme. La taille du FO, à une profondeur donnée, correspond à la disparité stéréoscopique entre les deux positions extrêmes de ces centres optiques sur l'axe horizontal. Le modèle de FO a seulement deux paramètres : la *distance de mise au point* (DMP) et la *profondeur de champ* (PC).

On peut considérer qu'un élément de la scène est *net* s'il est à une disparité qui a une taille de FO plus petite qu'un seuil fixe. Les éléments à une disparité avec une taille de

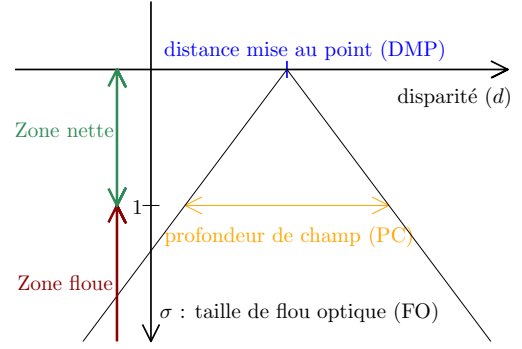


FIG. 1 – Le modèle de flou dû à l'optique : la taille du FO est une fonction de la disparité stéréoscopique linéaire par morceaux. Elle est nulle à la disparité correspondante à la distance de mise au point (DMP) et sa pente est inverse à la profondeur de champ (PC).

FO plus grande seront considérés comme *flous*.

2.2 Différence de deux modèles de flou dû à l'optique

Si nous avons deux fonctions de FO, une par caméra, nous pouvons considérer leur différence. Chaque paramètre (DMP et PC) du modèle, peut être rangé en trois catégories : il peut être plus grand à gauche qu'à droite, égal, ou plus petit à gauche qu'à droite. Une translation globale ou un facteur d'échelle appliqués aux paramètres ne changera pas la forme de la courbe. Sur le tableau de la Fig. 2 nous représentons graphiquement les neuf cas, issus des combinaisons de trois possibilités pour chaque paramètre.

2.3 Flou dans l'image

Nous venons d'introduire le modèle de FO. Mais dans notre méthode, nous avons seulement les images en entrée, et pas de connaissance des réglages des caméras. Ceci implique que nous ne pouvons pas mesurer le FO directement, mais seulement le *flou dans l'image* (FI).

Même si nous ne pouvons pas mesurer le FO à partir des

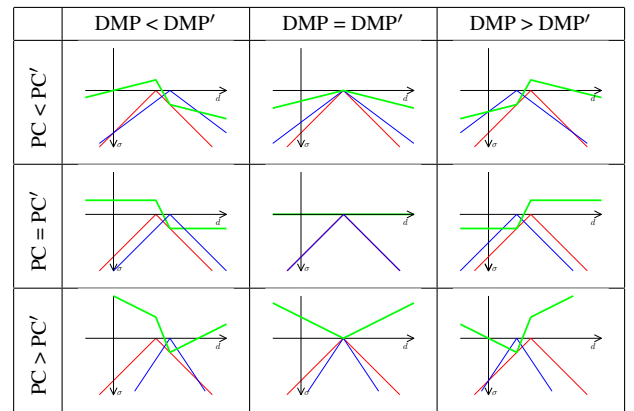


FIG. 2 – Graphiques de flou dû à l'optique et leur différence. Chaque case contient les graphiques des fonctions gauche en rouge, droite en bleu, puis leur différence en vert. Les paramètres de l'image gauche sont DMP et PC. DMP' et PC' sont les paramètres de l'image droite.

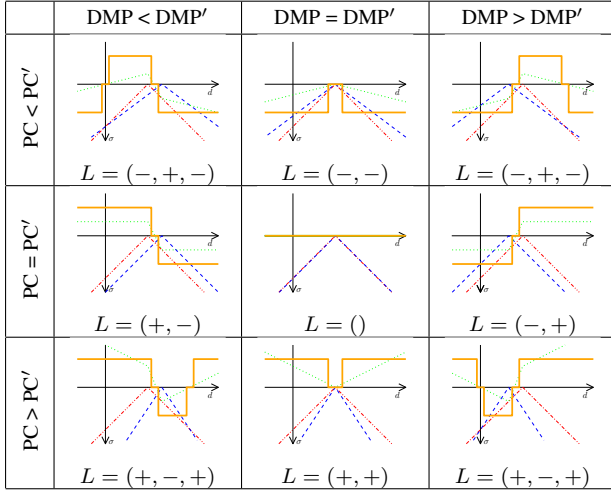


FIG. 3 – *Catalogue* des types de courbes possibles pour chaque configuration. En orange, le signe de la différence de FO. En pointillées les fonctions de FO gauche (rouge), FO droite (bleu) et leur différence (vert). Pour chaque courbe nous donnons aussi la liste L de signes que nous utiliserons dans la méthode de classification (Sec 3.5).

images, un des éléments clés de notre méthode est de supposer que, pour chaque pixel, la différence de FI a le même signe que la différence de FO. Nous allons, donc, utiliser le signe de la différence de FI pour détecter une différence dans le FO. La Fig. 3 montre les graphiques correspondant au signe du FO en orange. Bien que le flou dans l'image dépende aussi de la texture de la scène, la courbe du signe de la différence de FI devrait être égal à la courbe du signe de la différence de FO. Cet ensemble de courbes constitue le *catalogue* de formes admissibles pour les graphiques de signe du FI.

Notons que si nous considérons seulement la forme des courbes de la Fig. 3, nous sommes incapables de différencier les cas (PC < PC', DMP < DMP') et (PC < PC', DMP > DMP'). De façon analogue (PC > PC', DMP < DMP') a la même courbe que (PC > PC', DMP > DMP'). Dans la Sec. 3.5, nous explicitons sous quelles conditions nous pouvons obtenir des informations utiles dans ces cas-là.

3 Mesure de différence de mise au point

Nous détaillons ci-après l'algorithme pour calculer la différence de MAP entre les images gauche et droite, exprimée en termes de sa DMP et de sa PC. Nous commençons par calculer une carte dense de disparité qui donne les correspondances entre les pixels des images, en utilisant une méthode multi-résolution peu sensible à la différence de flou [13]. Ensuite, nous mesurons le FI et le signe de cette différence pour chaque paire de pixels mis en relation (par la carte de disparité). Nous estimons sur ces mesures bruitées un modèle de *différence de FO*. Finalement, nous essayons d'extraire de l'information utile à partir du modèle estimé et de donner un retour à l'opérateur. Les étapes de l'algorithme sont :

1. Calculer une carte de disparité dense ;
2. Mesurer la différence de flou dans l'image pour chaque pixel, et additionner ces valeurs par disparité ;
3. Estimer un modèle sur ces mesures ;
4. Classifier le modèle obtenu ;
5. Donner un retour à l'opérateur.

3.1 Calcul des cartes de disparité

D'abord nous avons besoin de trouver la disparité stéréoscopique qui relie les images gauche et droite. Le calcul de la disparité stéréoscopique est en lui-même un sujet de recherche dans le domaine de la vision par ordinateur. Des avancées récentes montrent que, dans la plupart des cas, des disparités précises peuvent être calculées rapidement, voire à cadence vidéo. Par contre, dans des situations difficiles, comme une profondeur de champ réduite, des zones manquant de texture, des discontinuités de profondeur, des patrons répétitifs, des transparences ou des réflexions spéculaires, la disparité reste encore très difficile à estimer. Ces situations sont souvent la cause d'erreurs locales dans la plupart des méthodes.

Nous pouvons supposer que les images originales sont rectifiées. Elles n'ont pas de disparité verticale : un point (x, y) dans l'image gauche correspond à un point sur la même ordonnée y , ayant des coordonnées (x', y) dans l'image droite.

Pour notre problématique, nous avons besoin d'une méthode de calcul de disparité rapide qui se comporte bien dans le cas de PC réduite. Nous utilisons donc la méthode [13] capable de trouver des bonnes valeurs de disparité entre une image nette et une autre floue. Cette méthode n'est pas la seule de l'état de l'art qui remplisse nos contraintes et nous pourrions en utiliser d'autres.

D'autre part, nous voulons ignorer les éléments des images qui sont demi-occlus, puisqu'ils n'ont pas de correspondant sur l'autre image. Afin de détecter ces zones, nous considérons $d(x, y)$, la disparité gauche-droite du pixel $p = (x, y)$. Ce pixel correspond aussi au pixel $(x', y) = (x + d(x, y), y)$ sur l'image droite. Si les disparités ne sont pas proches ($|d(x, y) - d'(x', y)| < D_T$), un pixel en cache un autre et nous éliminerons du processus le pixel le plus lointain. Typiquement, la valeur de D_T est 1.

3.2 Mesure du flou dans l'image

Pour comparer le flou dans l'image entre des points correspondants sur les images, nous utilisons l'opérateur SML (*Sum of Modified Laplacian* [2, 8]). Cet opérateur fût conçu pour des applications de *depth-from-focus* :

$$\begin{aligned}
 \text{SML}(p) &= \sum_{i=x-N}^{x+N} \sum_{j=y-N}^{y+N} \nabla_{ML}^2 I(i, j), \\
 &\text{pour } \nabla_{ML}^2 I(i, j) \geq T, \text{ où} \\
 \nabla_{ML}^2 I(x, y) &= |2I(x, y) - I(x - s, y) - I(x + s, y)| \\
 &\quad + |2I(x, y) - I(x, y - s) - I(x, y + s)|
 \end{aligned}$$

T est un seuil discriminatif, N est la taille de la fenêtre utilisée pour le calcul du SML, et s est la taille du pas pour le calcul du Laplacien. Dans nos expérimentations les paramètres étaient $T = 5$ (images 8-bits), $N = 1$ et $s = 1$. Pour un pixel $p = (x, y)$ dans l'image *gauche*, $SML(p)$ est sa réponse à l'opérateur SML. $SML'(p)$ est la réponse de l'opérateur sur son pixel correspondant sur l'image *droite*. Nous écrirons $S(p)$ pour désigner le signe de la différence des SML entre deux pixels :

$$S(p) = \text{signe} (SML(p) - SML'(p)) \quad (1)$$

Elle est positive si l'image gauche est plus nette que la droite, et elle est négative si la droite est plus nette que la gauche. Elle est nulle si $SML(p) = SML'(p)$. Typiquement, des zones sans texture ont des valeurs SML nulles.

Notons qu'une différence importante de contraste entre les images peut fortement perturber ces mesures. Une zone floue et très contrastée risque d'être considérée comme plus nette qu'une zone nette mais peu contrastée. Néanmoins la différence de contraste entre les images d'une paire stéréoscopique peut être calculée facilement, puis corrigée par une transformation linéaire des valeurs de l'image moins contrastée. Des différences locales de contraste peuvent subsister, introduisant un léger biais sur les mesures.

Pour avoir une indication sur la fiabilité de la différence de flou dans les images, nous définissons :

$$m(p) = \max (|SML(p)|, |SML'(p)|)$$

Cette valeur sera élevée si l'image est texturée et nette.

Pour chaque valeur de disparité dans la scène, $M(d)$ sera la moyenne des $S(p)$ pour les pixels qui sont à disparité d , pondérés par $m(p)$. $w(d)$ sera la somme de poids correspondante. Donc

$$M(d) = \frac{\sum_{p|d(p)=d} m(p)S(p)}{\sum_{p|d(p)=d} m(p)}$$

et

$$w(d) = \sum_{p|d(p)=d} m(p)$$

Ces éléments nous donnent une estimation de quelle image est plus nette pour les pixels de même disparité.

Évidemment cette estimation peut être bruitée. Les sources d'erreur les plus importantes que nous avons déjà évoquées sont :

- La valeur de disparité peut être peu précise, impliquant une erreur de mise en correspondance entre (x, y) et (x', y) . $S(p)$ risque d'être faux.
- Des différences locales de contraste peuvent biaiser $S(p)$, même si la disparité est bien estimée.
- Le FO dans des zones de l'avant-plan peut «baver» à travers les discontinuités de profondeur : des zones de l'arrière-plan peuvent être mesurées à tort comme floues.

- Le nombre de pixels disponibles dans chaque disparité dépend du contenu de la scène. L'information pour certaines valeurs de profondeur peut être inexistante.

Pour compenser ces erreurs nous utiliserons des méthodes robustes pour l'estimation du modèle.

3.3 Estimation du modèle de flou dû à l'optique à partir du signe de la différence de flou dans l'image

Bien que les mesures soient bruitées, le signe de la différence de FI, en fonction de la disparité, doit avoir une des formes présentées dans la Fig. 3. Dans cette étape nous estimons un modèle qui respecte ce *catalogue* (Fig. 3).

En entrée nous avons les mesures $M(d)$ et $w(d)$ et nous cherchons une fonction $C(d)$ qui minimise l'énergie :

$$E(C) = \sum_d w(d)E_{\text{données}}(d) + \lambda E_{\text{régul}}(d), \text{ avec } \begin{cases} E_{\text{données}}(d) = |M(d) - C(d)| \\ E_{\text{régul}}(d) = |C(d-1) - C(d)| \end{cases}$$

$E_{\text{données}}$ est le terme d'attache aux données. Comme le modèle que nous cherchons est continu par morceaux en d , nous introduisons le terme $E_{\text{régul}}$ comme régularisateur. Pour chaque disparité d , la fonction $C(d)$ peut prendre un des états suivants : “Gauche plus nette que droite”, “Droite plus nette que gauche” et “Les deux images ont la même netteté”. Nous représentons ces états en valeurs numériques $+1, 0, -1$. Nous voulons que notre implantation du calcul soit robuste au bruit, donc nous permettons à $C(d)$ de prendre 5 valeurs possibles :

- Droite plus nette (-0.7)
- Droite peut-être plus nette (-0.3)
- Même netteté (0)
- Gauche peut-être plus nette (0.3)
- Gauche plus nette (0.7)

Le fait que la courbe $C(d)$ doive prendre une des formes correspondantes à la Fig. 3 nous rajoute une dernière contrainte : le nombre de changements de signe est limité. Pour chaque passage d'une valeur positive ou négative vers zéro, nous additionnons une transition. Par contre si on passe d'une valeur positive à une négative directement (ou l'inverse), nous comptons deux transitions. En observant les formes de la Fig. 3 on peut voir que le nombre maximal de transitions permis est de quatre.

Néanmoins, cette condition est nécessaire mais pas suffisante. Certaines courbes ont quatre transitions ou moins mais ne correspondent à aucune courbe de notre *catalogue*. Ces courbes ne correspondent pas non plus à une partie (un sous-ensemble) de nos modèles (plus d'information sur les modèles incomplets dans la Sec. 3.5). La Fig. 4 montre des formes de courbes qui sont interdites. Ces formes interdites seront éliminées de nos candidates pendant l'optimisation. Nous pouvons résoudre cette optimisation en temps linéaire ($O(d_{\text{max}} - d_{\text{min}})$) en utilisant la programmation dynamique. Le résultat est une courbe qui, pour chaque dis-

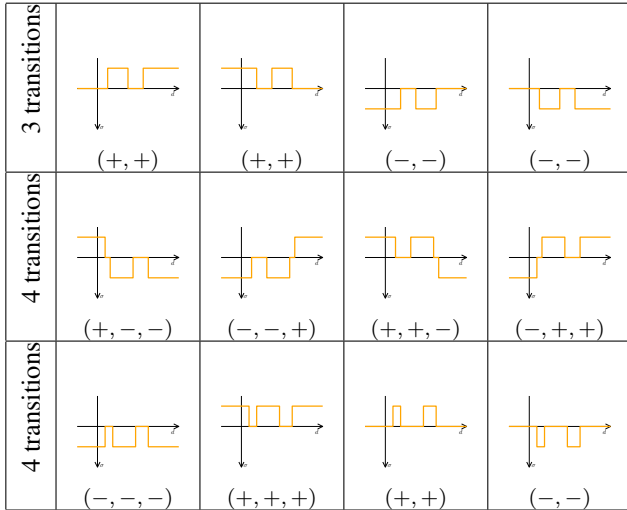


FIG. 4 – Courbes interdites : elles ne correspondent pas aux courbes du *catalogue* ni à des sous-ensembles. Pour chaque courbe nous donnons aussi la liste des signes utilisés dans l'étape de classification (voir Sec. 3.5).

parité, nous dit si la gauche est plus nette, si les deux ont la même netteté ou si la droite est plus nette.

3.4 Visualisation de la différence de MAP

A cette étape nous sommes déjà capables de visualiser le résultat brut de notre méthode. Nous proposons de dessiner des zébrures sur les zones de l'image qui sont moins nettes que les autres. Nous pourrions faire ce dessin par pixel, c.-à-d. en utilisant l'information de $S(p)$ de l'Eq. 1, mais nous nous sommes aperçus que le résultat était trop bruité et difficile à interpréter (voir Fig. 7).

Nous pouvons obtenir une bien meilleure visualisation en utilisant la fonction $C(d)$. Nous avons la disparité d pour chaque pixel, et nous pouvons dessiner les zébrures à partir du modèle $C(d)$ obtenu. Nous dessinons des zébrures dans un sens si une image est *peut être moins nette* (c.-à-d. $C(d) = -0.3$ pour l'image gauche). Et nous dessinons des zébrures dans les deux sens si nous sommes sûrs que l'image est *moins nette* (c.-à-d. $C(d) = -0.7$ toujours pour l'image gauche). De cette façon nous évitons de montrer le bruit présent dans la mesure $S(p)$, et nous pouvons interpréter facilement les images obtenues (Fig. 9).

Nous donnons donc une information visuelle sur quelle caméra devrait être réglée, et nous pouvons laisser la liberté à l'opérateur de décider quelle est la cause de cette différence. Dans la plupart des cas, spécialement s'il y a assez de mesures autour de la zone de DMP des deux caméras, nous sommes capables de classifier $C(d)$ et donner des indications plus précises sur la nature du problème.

3.5 Classification de la fonction du signe de la différence de flou des images

Nous voulons classer la fonction $C(d)$ dans une des cases de notre *catalogue* de la Fig. 3. Pour cela, nous construisons une liste L constituée des parties positives et négatives

de la fonction, ordonnées par disparité croissante. Dans la Fig. 3 et Fig. 4 nous montrons, pour chaque forme, sa liste L . Une fonction $C(d)$ autorisée peut avoir un maximum de quatre transitions de signe, donc notre liste n'aura au maximum que trois éléments, et aucun au minimum.

Cas où des mesures sont disponibles pour toutes les disparités. Si nous avons des mesures pour *toutes les valeurs de disparité*, correspondante aux profondeurs de zéro à l'infini, nous pourrions classer facilement la fonction $C(d)$ en utilisant la liste L et notre *catalogue*. Comme nous avons rejeté les configurations interdites dans les étapes précédentes, les seules configurations possibles sont :

$$L \in \{(), (+), (-), (+, +), (-, -), (+, -), (-, +), (+, -, +), (-, +, -)\}$$

Chaque configuration, exceptées $L = (+)$ et $L = (-)$, peut être mise en relation avec un élément de la Fig. 3 et nous donner des indications sur la cause du problème.

Néanmoins, la supposition que toutes les mesures sont disponibles pour toutes les disparités n'est pas très réaliste. Nous allons donc revoir cette supposition.

Un cas plus réaliste : des mesures clairsemées. Selon la géométrie de la scène, nous n'aurons de l'information que sur quelques valeurs de disparité. Ces mesures nous donneront une vision partielle du modèle complet de la différence de flou. Dans quelques cas, des visions partielles de modèles différents donneront lieu à la même mesure, ce qui rendra difficile le choix du modèle qui a généré les mesures.

Nous présentons un exemple, montrant la difficulté d'estimer le bon modèle à partir de mesures partielles.

$$\text{Soit } C(d) = \begin{cases} +1 & \text{si } d \leq d_0, \\ -1 & \text{si } d > d_0. \end{cases} \text{ et } L = (+, -).$$

Les modèles candidats sont montrés dans la Fig. 5a.

Dans cet exemple, un grand ensemble de modèles candidats correspondent à notre fonction $C(d)$. Pour réduire leur nombre, une supposition raisonnable pourrait être de considérer que l'opérateur a fait le point d'une caméra sur un objet de la scène, et que l'autre DMP est aussi dans l'intervalle de disparité observable. Cette hypothèse réduit le nombre de candidats possibles pour notre modèle. Il ne nous reste que les candidats de la Fig. 5b, qui partagent la relation $\text{DMP} < \text{DMP}'$.

Nous pouvons affirmer que si nous avons des mesures sur les disparités correspondantes aux DMP, nous pouvons *toujours* répondre à la question : *quelle distance de mise au point est la plus proche ?* Une fois que l'opérateur a corrigé les distances de mise au point ($\text{DMP} = \text{DMP}'$), nous pouvons *toujours* répondre à la question : *quelle profondeur de champ est la plus grande ?*

4 Résultats

Cette section est composée de trois parties. D'abord nous expliquons la génération d'images de test pour valider la

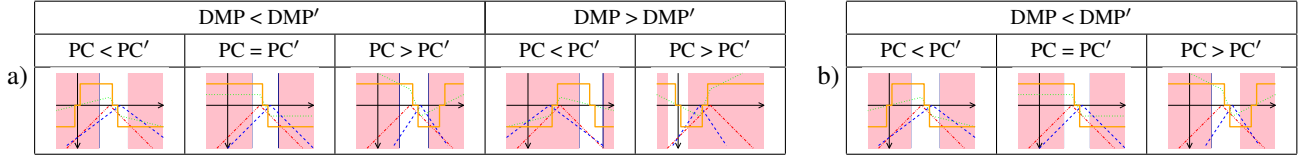


FIG. 5 – a) Candidats possibles pour l’exemple où $C(d)$ change de positive à négative dans la partie visible de la disparité (intervalle non-rayé) : 5 configurations possibles. b) Candidats possibles si on suppose que les disparités correspondantes aux DMP des deux caméras sont dans l’intervalle de disparité mesurable (intervalle non-rayé) : plus que 3 configurations possibles.

méthode. Ensuite, nous illustrons les étapes de la méthode avec un exemple. Finalement, nous analysons les résultats obtenus avec la méthode et le jeu d’images générées.

4.1 Images synthétiques générées par lancer de rayons

Pour valider notre méthode, nous avons généré un jeu d’images de test¹. Pour avoir un contrôle total sur les paramètres de la caméra nous avons utilisé un moteur de rendu qui utilise le lancer de rayons (POV-Ray²) et nous avons généré des images de la scène “Patio” de Jaime Vives Piqueres³ avec une caméra de modèle perspective. POV-Ray peut simuler le FO en calculant l’ensemble de rayons optiques voisins du pixel rendu en faisant sa moyenne. La taille de ce voisinage correspond à la taille du diaphragme. Ce modèle de génération de flou est donc similaire au nôtre (Sec. 2.1).

Nous avons généré des images avec des paramètres différents de PC et de DMP. Pour chaque vue (gauche et droite) nous avons généré 5 images : *Tout*, *Près*, *Près2*, *Loin* et *Loin2*. Les images *Tout* ont une PC infinie. Les images *Loin** ont une DMP à $d_L = 14.2$ (point vert sur la Fig. 7.a) et les images *Près** ont une DMP à $d_P = 10.2$ (point bleu sur la Fig. 7.a). Les images *Loin* et *Près* ont un diaphragme de valeur 5, qui génère une profondeur de champ finie. Les images *Loin2* et *Près2* ont un diaphragme de valeur 10 (double de *Loin* et *Près*), ayant donc une PC plus réduite.

4.2 Résultats Intermédiaires

Nous allons dérouler l’exemple (*GLoin, DPrès*). Dans la Fig. 6 nous avons un détail de ces images : à gauche la fontaine est floue, tandis qu’à l’arrière, les fleurs sont nettes. À droite la fontaine est nette, tandis que les fleurs sont floues. Dans la Fig. 7 nous observons $d(p)$: la carte de disparité calculée ; $S(p)$: le signe de la différence de SML ; et $m(p)$: le maximum de SML et SML'. L’image de $m(p)$ a été égalisée afin de montrer les détails dans le milieu de l’échelle de valeurs. Les mesures $S(p)$ sont, comme nous l’attendions, très bruitées. Mais une dominance du signe est visible pour les éléments à disparité d_P et d_L . Ces valeurs correspondent aux points bleus et verts sur la carte de disparité (Fig. 7a). L’image $m(p)$ montre quelle information est plus fiable : elle correspond aux zones texturées et nettes sur une image. Dans la Fig. 8 nous pouvons



FIG. 6 – Détail de la paire (*GLoin, DPrès*).

voir les données que l’étape d’estimation de modèle prend (Sec. 3.3) : $M(d)$, $w(d)$ et $C(d)$. Les valeurs minimale et maximale de $C(d)$ obtenues sont très proches des valeurs correctes des distances de mises au point. Ces valeurs indiquent un fort décalage entre les mises au point des caméras. Nous pouvons donc facilement répondre à la question la plus importante : *Y a-t-il une différence de MAP ?* Avec le modèle estimé et le *catalogue* nous pouvons conclure que $DMP > DMP'$ et $PC = PC'$. Éventuellement, $PC = PC'$ pourrait ne pas être vrai si nous n’avions qu’une vision partielle du modèle (intervalle de disparité réduit). $PC > PC'$ ou $PC < PC'$ sont aussi possibles.

Finalement, dans la Fig. 9 nous montrons les zébrures obtenues sur les images originales (*GLoin, DPrès*) (la gauche avec mise au point sur l’arbre à l’arrière, la droite avec la mise au point sur la fontaine). Notons que les bonnes zones de chaque image ont été correctement identifiées.

4.3 Discussion

Afin de valider notre méthode nous l’avons testée avec des cartes de disparité vérité terrain et des cartes calculées.

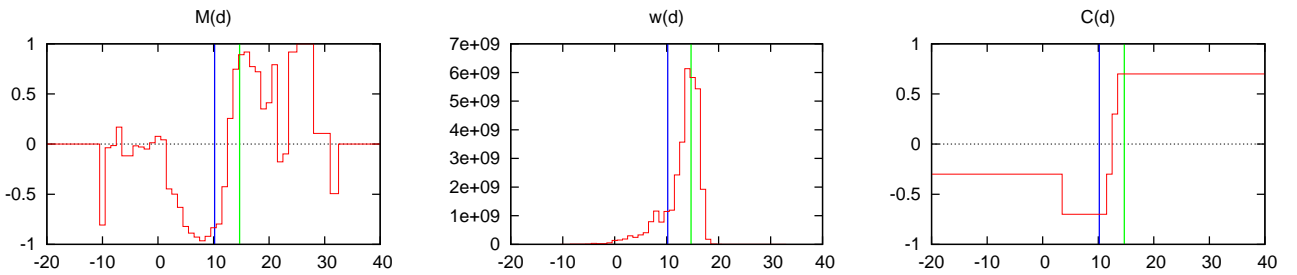
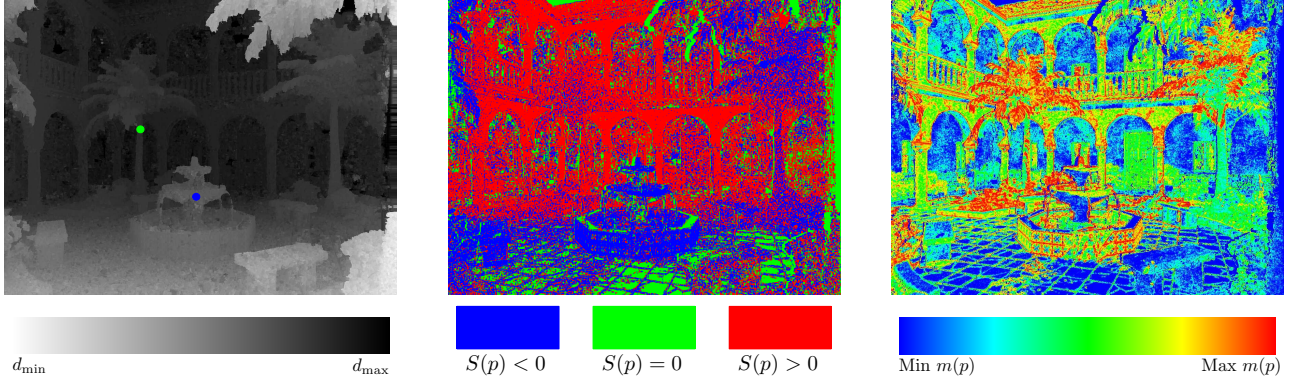


FIG. 9 – Images originales (*GLoin, DPrès*) avec les zébrures : la zone sur l’image gauche (c.-à-d. la fontaine) est moins nette. La zone sur l’image droite (c.-à-d. le fond) est moins nette.

¹<http://devernay.free.fr/vision/focus/patio>

²www.povray.org

³www.ignorancia.org



Nous avons réalisé les calculs entre chaque paire possible formée par nos 5 images originales. Nous avons donc 25 résultats que nous présentons dans les Fig. 10 et 11. Nous aimerions répondre aux questions suivantes, qui sont ordonnées par difficulté :

1. Les DMP et PC sont-elles identiques ?
2. Les DMP sont-elles identiques ? Laquelle est la plus grande ?
3. Les PC sont-elles identiques ? Laquelle est la plus grande ?

Dans tous les cas, nous pouvons répondre correctement à la première question : les éléments de la diagonale valent zéro, et les autres ont des valeurs non nulles.

Les quatre graphiques en bas à gauche et les autres quatre en haut à droite de la Fig. 10 montrent que nous pouvons répondre avec précision à la deuxième question quand les DMP sont différentes (c.-à-d. *Près* vs. *Loin* ou *Loin* vs. *Près*). Pour les quatre premiers, nous obtenons la même forme que dans l'exemple de la Sec. 4.2, et nous déduisons que $DMP > DMP'$. Pour le deuxième groupe, nous obtenons les courbes symétriques sur l'axe y , ce qui nous permet de conclure que $DMP < DMP'$. Par contre, dans tous ces huit cas, nous sommes incapables de répondre à la troisième question et d'affirmer s'il y a une différence de PC entre les caméras (voir Sec. 4.2).

Les douze graphiques restants correspondent à une différence de PC entre les caméras. Nous pouvons observer que

seulement deux d'entre eux ont la forme que notre modèle nous avait prédit. Pour ces cas, nous répondons correctement à toutes les questions : (GLoin, DLoin2) et (GLoin2, DLoin). Dans les dix restants, bien que nous ne puissions pas dire si la PC est différente à partir des courbes, nous pouvons toujours répondre à la question *Quelle image est la plus nette ?*

Lors que nous utilisons les disparités calculées (Fig. 11), nous observons que les résultats sont très similaires à ceux obtenus en utilisant la disparité vérité terrain. Notons qu'il n'y a pas de différence significative entre les Fig. 10 et Fig. 11.

	DPrès2	DPrès	DTout	DLoin	DLoin2
GPrès2					
GPrès					
GTout					
GLoin					
GLoin2					

FIG. 10 — Modèles estimés avec la disparité vérité terrain.

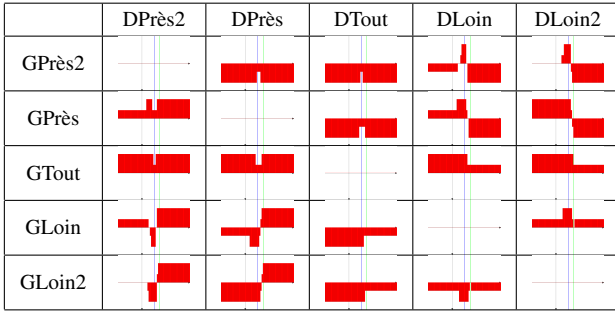


FIG. 11 – Modèles estimés avec la disparité *calculée*.

5 Conclusions et perspectives

Nous avons présenté une méthode complète pour détecter une différence de MAP entre une paire d'images stéréoscopiques. Comme il est impossible de récupérer l'information perdue après le tournage, il est très important que les DMP et les PC soient identiques.

Les résultats obtenus sur les images générées par lancer de rayons sont très prometteurs. Tous les éléments de la diagonale de la Fig. 11 valent zéro, ce qui veut dire que notre méthode est très précise pour le cas le plus important : les deux réglages sont identiques. Un autre point positif est que nous n'avons aucun *faux positif* : toutes les configurations présentant un problème sont détectées correctement. Nous pouvons affirmer que notre méthode a un taux de réussite de 100% lorsque l'on veut répondre à la question : *Les DMP et les PC sont-elles identiques ?* Dans les configurations où les DMP sont différentes, notre méthode les détecte correctement et nous sommes capables de dire quelle DMP est plus petite. Dans le cas où il y a une différence de PC, nous n'arrivons pas toujours à la détecter. Néanmoins nous pouvons toujours indiquer quelle caméra est la plus nette.

Nos résultats obtenus avec les disparités *vérité terrain* et les disparités *calculées* sont très similaires. Nous pouvons donc affirmer que la méthode choisie pour le calcul des disparités s'adapte à notre besoin (images floues en entrée). Toutes les étapes de la méthode peuvent être calculées en temps-réel : les cartes de disparité, la différence de SML, l'estimation du modèle, sa classification et le retour en forme de zébrures. La méthode finale pourrait donc être implantée en temps-réel et intégrée dans des outils de tournage pour aider les opérateurs à résoudre les différences de MAP sur le terrain.

Les travaux futurs devraient viser la validation de la méthode sur des images réelles, ainsi que la validation de la part des opérateurs sur le retour en forme de zébrures.

Dans la profession de cinéaste, il est connu que l'utilisation d'un miroir séparateur d'images peut introduire d'autres déformations optiques. Par exemple, si le miroir n'est pas parfaitement plat, nous pouvons observer un flou anisotrope, correspondant à l'astigmatisme optique. De futures recherches pourraient se concentrer sur une meilleure caractérisation du FO pour prendre en compte ces aberrations.

tions.

Références

- [1] FitzGerrell, A. R., Edward R. Dowski, j., and Cathey, W. T., *Defocus transfer function for circularly symmetric pupils* Appl. Opt. 36, 5796–5804 (Aug. 1997).
- [2] Huang, W. and Jing, Z., *Evaluation of focus measures in multi-focus image fusion*, Pattern Recognition Letters 28(4), 493–500 (2007).
- [3] Frank L. Kooi and Alexander Toet, *Visual comfort of binocular and 3D displays* Displays 25, 99–108 (Aug. 2004).
- [4] Kroon, R. W., *3D A to Z*, in [3D TV and 3D Cinema], Mendiburu, B., ed., 195 – 232, Focal Press, Boston (2012).
- [5] Marc T. M. Lambooi and Wijnand A. Ijsselstein and Ingrid Heynderickx, *Visual discomfort in stereoscopic displays : a review in* [Proc. SPIE Stereoscopic Displays and Virtual Reality Systems XIV], 6490 2007.
- [6] Lin, H.-Y. and Gu, K.-D., *Photo-realistic depth-of-field effects synthesis based on real camera parameters*, in [Advances in Visual Computing (ISVC 2007)], Lecture Notes in Computer Science 4841, 298–309, Springer-Verlag (Nov. 2007).
- [7] Mennucci, A. and Soatto, S., *On observing shape from defocused images*, in [Image Analysis and Processing, 1999. Proceedings. International Conference on], 550 –555 (1999).
- [8] Nayar, S. and Nakagawa, Y., *Shape from focus*, IEEE Trans. on Pattern Analysis and Machine Intelligence 16, 824–831 (Aug. 1994).
- [9] Pentland, A. P., *A new sense for depth of field* IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 9, 523 –531 (July 1987).
- [10] Rajagopalan, A., Chaudhuri, S., and Mudénagudi, U., *Depth estimation and image restoration using defocused stereo pairs*, IEEE Trans. on Pattern Analysis and Machine Intelligence 26, 1521 –1525 (Nov. 2004).
- [11] Schechner, Y. and Kiryati, N., *Depth from defocus vs. stereo : how different really are they ?* in [Proc. Intl. Conf. on Pattern Recognition], 2, 1784–1786 (Aug. 1998).
- [12] Seuntjens, P., Meesters, L., and Ijsselstein, W., *Perceived quality of compressed stereoscopic images : Effects of symmetric and asymmetric JPEG coding and camera separation*, ACM Trans. Appl. Percept. 3, 95–109 (Apr. 2009).
- [13] Sizintsev, M. and Wildes, R. P., *Coarse-to-fine stereo vision with accurate 3D boundaries*, Image and Vision Computing 28(3), 352 – 366 (2010).
- [14] Stelmach, L., Tam, W., Meegan, D., Vincent, A., and Corribeau, P., *Human perception of mismatched stereoscopic 3D inputs*, in [International Conference on Image Processing (ICIP)], 1, 5–8 (2000).
- [15] Subbarao, M., *Parallel depth recovery by changing camera parameters*, in [Intl. Conf. on Computer Vision.], 149–155 (Dec. 1988).
- [16] Subbarao, M. and Surya, G., *Depth from defocus : A spatial domain approach* International Journal of Computer Vision 13, 271–294 (1994).